REPORT DOCUMENTATION PAGE

Form Approved OMB NO. 0704-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggesstions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA, 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any oenalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE (DD-MM-YYYY) 15-07-2014	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 1-Jun-2009 - 30-Apr-2013

4. TITLE AND SUBTITLE
Final Report: Density Estimation and Anomaly Detection in Large Social Networks

5a. CONTRACT NUMBER
W911NF-09-1-0262

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
611102

6. AUTHORS
Rebecca Willett

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES
Duke University
2200 W Main St Ste 710

Durham, NC 27705 -4677

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES)
U.S. Army Research Office
P.O. Box 12211
Research Triangle Park, NC 27709-2211

10. SPONSOR/MONITOR'S ACRONYM(S)
ARO

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
54740-MA.16

12. DISTRIBUTION AVAILIBILITY STATEMENT
Approved for Public Release; Distribution Unlimited

13. SUPPLEMENTARY NOTES
The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

High-velocity streams of high-dimensional data pose significant “big data” analysis challenges across a range of applications and settings, including large-scale social network analysis. Online learning and online convex programming play a significant role in the rapid recovery of important or anomalous information from these large data streams. While recent advances in online learning have led to novel and rapidly converging algorithms, these methods are unable to adapt to non-stationary environments arising in real-world settings. This paper describes a dynamic mirror descent framework which addresses this challenge, yielding low theoretical regrets bounds and

15. SUBJECT TERMS
Online learning, social networks, dynamical models, big data

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Rebecca Willett
a. REPORT UU	b. ABSTRACT UU	c. THIS PAGE UU	UU		19b. TELEPHONE NUMBER 608-316-4357

Report Title

Final Report: Density Estimation and Anomaly Detection in Large Social Networks

ABSTRACT

High-velocity streams of high-dimensional data pose significant “big data” analysis challenges across a range of applications and settings, including large-scale social network analysis. Online learning and online convex programming play a significant role in the rapid recovery of important or anomalous information from these large data streams. While recent advances in online learning have led to novel and rapidly converging algorithms, these methods are unable to adapt to non-stationary environments arising in real-world settings. This paper describes a dynamic mirror descent framework which addresses this challenge, yielding low theoretical regrets bounds and accurate, adaptive, and computationally efficient algorithms which are applicable to broad classes of problems. The methods are capable of learning and adapting to the underlying and possibly time-varying dynamics of a system or environment. Empirical results in the context of social network tracking, dynamic texture analysis, sequential compressed sensing of a dynamic scene, and tracking self-exciting point processes support the core theoretical findings.

Enter List of papers submitted or published that acknowledge ARO support from the start of the project to the date of this printing. List the papers, including journal references, in the following categories:

(a) Papers published in peer-reviewed journals (N/A for none)

Received	Paper
07/15/2014 13.00	Maxim Raginsky, Jorge G. Silva, Svetlana Lazebnik, Rebecca Willett. A recursive procedure for density estimation on the binary hypercube, Electronic Journal of Statistics, (03 2013): 820. doi:
07/15/2014 14.00	Maxim Raginsky, Rebecca M. Willett, Corinne Horn, Jorge Silva, Roummel F. Marcia. Sequential Anomaly Detection in the Presence of Noise and Limited Feedback, IEEE Transactions on Information Theory, (08 2012): 5544. doi: 10.1109/TIT.2012.2201375
TOTAL:	**2**

Number of Papers published in peer-reviewed journals:

(b) Papers published in non-peer-reviewed journals (N/A for none)

Received	Paper
TOTAL:	

Number of Papers published in non peer-reviewed journals:

(c) Presentations

1. Workshop on “Eigenvectors in graph theory and related problems in numerical linear algebra” at Institute for Computational and Experimental Research in Mathematics (ICERM) at Brown University, May 2014.

2. Information Theory and Applications at UC San Diego, Feb. 2014.

3. Department seminar at UCLA, Jan. 2014.

4. Department seminar at Caltech, Jan. 2014.

5. Keynote at IEEE GlobalSIP workshop, Dec. 2013.

6. Presentation at SAMSI workshop on social networks.

7. Presentation at Sensing, Information, Learning and Optimization workshop at UW Madison, June 2013

8. Seminar at Janelia Farm, April 2013.

9. Seminar at University of Michigan, April 2013.

10. Seminar at Tampere University of Technology, Feb. 2013.

Number of Presentations: 10.00

Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received	Paper

TOTAL:

Number of Non Peer-Reviewed Conference Proceeding publications (other than abstracts):

Peer-Reviewed Conference Proceeding publications (other than abstracts):

Received		Paper
07/15/2014	15.00	Eric Hall, Rebecca Willett. Online Optimization in Parametric Dynamic Environments, Annual Allerton Conference on Communication, Control, and Computing. 01-OCT-12, . : ,
10/11/2011	2.00	Eric Wang, Jorge Silva, Rebecca Willett, Lawrence Carin. TIME-EVOLVING MODELING OF SOCIAL NETWORKS, ICASSP. 22-MAY-11, . : ,
10/11/2011	1.00	Rebecca Willett, Corinne Horn. Online anomaly detection with expert system feedback in social networks, ICASSP. 22-MAY-11, . : ,
10/11/2011	6.00	Yunchao Gong, Svetlana Lazebnik. Iterative Quantization: A Procrustean Approach to Learning Binary Codes, IEEE Conference on Computer Vision and Pattern Recognition. 20-JUN-11, . : ,
10/11/2011	7.00	Yunchao Gong, Svetlana Lazebnik. Comparing Data-Dependent and Data-Independent Embeddings for Classification and Ranking of Internet Images, IEEE Conference on Computer Vision and Pattern Recognition. 20-JUN-11, . : ,
10/13/2011	8.00	Eric Wang, Jorge Silva, Rebecca Willett, Lawrence Carin. DYNAMIC RELATIONAL TOPIC MODEL FOR SOCIAL NETWORK ANALYSIS WITH NOISY LINKS, Statistical Signal Processing Workshop. 28-JUN-11, . : ,

TOTAL: **6**

Number of Peer-Reviewed Conference Proceeding publications (other than abstracts):

(d) Manuscripts

Received		Paper
07/15/2014	12.00	Eric Hall, Rebecca Willett. Online Optimization in Dynamic Environments, IEEE Journal of Selected Topics in Signal Processing (06 2014)
10/13/2011	9.00	Maxim Raginsky, Rebecca Willett, Corinne Horn, Jorge Silva, Roummel Marcia. Sequential Anomaly Detection in the Presence of Noise and Limited Feedback, (submitted) (10 2011)

TOTAL: **2**

Number of Manuscripts:

Books

Received	Book

TOTAL:

Received	Book Chapter

TOTAL:

Patents Submitted

Patents Awarded

Awards

Keynote speaker at SMAI Curves and Surfaces Workshop
Plenary speaker at SIAM Conference on Imaging Science
Elected SIAM Imaging Science Program Director

Graduate Students

NAME	PERCENT SUPPORTED	Discipline
Eric Hall	1.00	
Zachary Harmany	0.10	
FTE Equivalent:	**1.10**	
Total Number:	**2**	

Names of Post Doctorates

NAME	PERCENT_SUPPORTED
Jorge Silva	0.20
Maxim Raginsky	0.20
FTE Equivalent:	**0.40**
Total Number:	**2**

Names of Faculty Supported

NAME	PERCENT_SUPPORTED	National Academy Member
Rebecca Willett	0.08	
FTE Equivalent:	**0.08**	
Total Number:	**1**	

Names of Under Graduate students supported

NAME	PERCENT_SUPPORTED	Discipline
Corinne Horn	0.00	Computer and Computational Science and Electrical E
FTE Equivalent:	**0.00**	
Total Number:	**1**	

Student Metrics

This section only applies to graduating undergraduates supported by this agreement in this reporting period

The number of undergraduates funded by this agreement who graduated during this period: 1.00

The number of undergraduates funded by this agreement who graduated during this period with a degree in science, mathematics, engineering, or technology fields:...... 1.00

The number of undergraduates funded by your agreement who graduated during this period and will continue to pursue a graduate or Ph.D. degree in science, mathematics, engineering, or technology fields:...... 1.00

Number of graduating undergraduates who achieved a 3.5 GPA to 4.0 (4.0 max scale):...... 1.00

Number of graduating undergraduates funded by a DoD funded Center of Excellence grant for Education, Research and Engineering:...... 0.00

The number of undergraduates funded by your agreement who graduated during this period and intend to work for the Department of Defense 1.00

The number of undergraduates funded by your agreement who graduated during this period and will receive scholarships or fellowships for further studies in science, mathematics, engineering or technology fields:...... 1.00

Names of Personnel receiving masters degrees

NAME	
Eric Hall	
Total Number:	**1**

Names of personnel receiving PHDs

NAME	
Zachary Harmany	
Total Number:	**1**

Names of other research staff

NAME	PERCENT_SUPPORTED
FTE Equivalent:	
Total Number:	

Sub Contractors (DD882)

1 a. Svetlana Lazebnik

1 b. 201 N Goodwin Ave

Urbana IL 61801

Sub Contractor Numbers (c):

Patent Clause Number (d-1):

Patent Date (d-2):

Work Description (e):

Sub Contract Award Date (f-1): 6/1/09 12:00AM

Sub Contract Est Completion Date(f-2): 4/30/13 12:00AM

Inventions (DD882)

Scientific Progress

See Attachment

Technology Transfer

Final Report for W911NF0910262

Contents

List of Figures

Statement of the problem studied

1.1 Introduction

Modern sensors are collecting very high-dimensional data at unprecedented volume and speed, often from platforms with limited processing power. These large datasets allow scientists and analysts to consider richer physical models with larger numbers of variables, and thereby have the potential to provide new insights into the underlying complex phenomena. Social media generate troves of data, but human analysts and machines cannot quickly and accurately identify inappropriate or dangerous content. The Large Hadron Collider (LHC) at CERN "generates so much data that scientists must discard the overwhelming majority of it – hoping hard they've not thrown away anything useful." [23] Typical NASA missions collect hundreds of terabytes of data every hour [18]: the Solar Data Observatory generates 1.5 terabytes of data daily [36], and the upcoming Square Kilometer Array (SKA, [19]) is projected to generate an exabyte of data daily, "more than twice the information sent around the internet on a daily basis and 100 times more information than the LHC produces" [35]. In these and a variety of other science and engineering settings, there is a pressing need to recover *relevant or anomalous* information *accurately and efficiently* from a high-dimensional, high-velocity data stream.

Rigorous analysis of such data poses major issues, however. First, we are faced with the notorious "curse of dimensionality", which states that the number of observations required for accurate inference in a stationary environment grows exponentially with the dimensionality of each observation. This requirement is often unsatisfied even in so-called "big data" settings, as the underlying environment varies over time in many applications. Furthermore, any viable method for processing massive data must be able to scale well to high data dimensions with limited memory and computational resources. Finally, in a variety of large-scale streaming data problems, ranging from motion imagery formation to network analysis, the underlying environment is dynamic yet predictable, but many general-purpose and computationally efficient methods for processing streaming data lack a principaled mechanism for incorporating dynamical models. Thus a fundamental mathematical and statistical challenge is accurate and efficient tracking of dynamic environments with high-dimensional streaming data.

Classical stochastic gradient descent methods, including the least mean squares (LMS) or recursive least squares (RLS) algorithms do not have a natural mechanism for incorporating dynamics. Classical stochastic filtering methods such as Kalman or particle filters or Bayesian updates [6] readily exploit dynamical models for effective prediction and tracking performance. However, these methods are also limited in their applicability because (a) they typically assume an accurate, fully known dynamical model and (b) they rely on strong assumptions regarding a generative model of the observations. Some techniques have been proposed to learn the dynamics [58, 53], but the underlying model still places heavy restrictions on the nature of the data. Performance analysis of these methods usually does not address the impact of "model mismatch", where the generative models are incorrectly specified.

A contrasting class of prediction methods, receiving widespread recent attention within the machine learning community, is based on an "individual sequence" or "universal prediction" [38] perspective; these strive to perform provably well on any individual observation sequence without assuming a generative model of the data. *Online convex programming* provides a variety of tools for sequential universal prediction [40, 8, 60, 17]. Here, a Forecaster measures its predictive performance according to a convex loss function, and with each new observation it computes the negative gradient of the loss and shifts its prediction in that direction. Stochastic gradient descent methods stem from similar principles and have been studied for decades, but recent technical breakthroughs allow these approaches to be understood without strong stochastic assumptions on the data, even in adversarial settings, leading to more efficient and rapidly converging algorithms in many settings.

This paper describes a novel framework for prediction in the individual sequence setting which incorporates dynamical models – effectively a novel combination of state updating from stochastic filter theory and online convex optimization from universal prediction. We establish tracking regret bounds for our proposed algorithm, *Dynamic Mirror Descent (DMD)*, which characterize how well we perform relative to some alternative approach (*e.g.,* a computationally intractable batch algorithm) operating on the same data to generate its own predictions, called a "comparator sequence." Our novel regret bounds scale with the deviation of this comparator sequence from a dynamical model. These bounds simplify to previously shown bounds when there are no dynamics. In addition, we describe methods based on DMD for adapting to the best dynamical model from either a finite or parametric class of candidate models. In these settings, we establish tracking regret bounds which scale with the deviation of a comparator sequence from the *best sequence* of dynamical models.

While our methods and theory apply in a broad range of settings, we are particularly interested in the setting where the dimensionality of the parameter to be estimated is very high. In this regime, the incorporation of both dynamical models and sparsity regularization plays a key role. With this in mind, we focus on a class of methods which incorporate regularization as well as dynamical modeling. The role of regularization, particularly sparsity regularization, is increasingly well understood in batch settings and has resulted in significant gains in ill-posed and data-starved settings [7, 45, 14, 9]. More recent work has examined the role of sparsity in online methods such as recursive least squares (RLS) algorithms, but do not account for dynamic environments [3].

1.1.1 Organization of paper and main contributions

The remainder of this paper is structured as follows. In Section 1.2, we formulate the problem and introduce notation used throughout the paper, and Section 1.3 provides some background definitions for online convex optimization. Our *Dynamic Mirror Descent (DMD)* method, along with tracking regret bounds are presented in Section 2.1; this section also describes the application of data-dependent dynamical models and their connection to recent work on online learning with predictable sequences. DMD uses only a single series of dynamical models, but we can use it to choose among a family of candidate dynamical models. This is described for finite families in Section 2.2 using a fixed share algorithm, and for parametric families in Section 2.3. Section 2.4 shows experimental results of our methods in a variety of contexts ranging from imaging to self-exciting point processes. Finally, Section 2.5 makes concluding remarks while proofs are relegated to Section **??**.

1.2 Problem formulation

The problem of sequential prediction is posed as an iterative game between a Forecaster and the Environment. At every time point, t, the Forecaster generates a prediction $\widehat{\theta}_t$ from a closed, convex set $\Theta \subset \mathbb{R}^d$. After the Forecaster makes a prediction, the Environment reveals the loss function

$\ell_t(\cdot)$ where ℓ_t is a convex function which maps the space Θ to the real number line. We will assume that the loss function is the composition of a convex function $f_t : \Theta \to \mathbb{R}$ from the Environment and a convex regularization function $r : \Theta \to \mathbb{R}$ which does not change over time. Frequently the loss function, f_t will measure the accuracy of a prediction compared to some new data point $x_t \in \mathsf{X}$ where X is the domain of possible observations. The regularization function promotes low-dimensional structure (such as sparsity) within the predictions. We additionally assume that we can compute a subgradient of ℓ_t or f_t at any point $\theta \in \Theta$, which we denote $\nabla \ell_t$ and ∇f_t. Thus the Forecaster incurs the loss $\ell_t(\widehat{\theta}_t) = f_t(\widehat{\theta}_t) + r(\widehat{\theta}_t)$.

The goal of the Forecaster is to create a sequence of predictions $\widehat{\theta}_1, \widehat{\theta}_2, \dots, \widehat{\theta}_T$ that has a low cumulative loss $\sum_t^T \ell_t(\widehat{\theta}_t)$. Because the loss functions are being revealed sequentially, the prediction at each time can only be a function of all previously revealed losses to ensure causality. Thus, the task facing the Forecaster is to create a new prediction, $\widehat{\theta}_{t+1}$, based on the previous prediction and the new loss function $\ell_t(\cdot)$, with the goal of minimizing loss at the next time step. We characterize the efficacy of $\widehat{\boldsymbol{\theta}}_T \triangleq (\widehat{\theta}_1, \widehat{\theta}_2, \dots, \widehat{\theta}_T) \in \Theta^T$ relative to a comparator sequence $\boldsymbol{\theta}_T \triangleq (\theta_1, \theta_2, \dots, \theta_T) \in \Theta^T$ using a concept called *regret*, which measures the difference of the total accumulated loss of the Forecaster with the total accumulated loss of the comparator. We are particularly interested in comparators which are computationally intractable batch algorithms; in a sense, then, regret encapsulates how much one regrets working in an online setting as opposed to a batch setting with full knowledge of past and future observations:

Definition 1 (Regret)**.** *The* regret *of $\widehat{\boldsymbol{\theta}}_T$ with respect to a comparator $\boldsymbol{\theta}_T \in \Theta^T$ is*

$$R_T(\boldsymbol{\theta}_T) \triangleq \sum_{t=1}^{T} \ell_t(\widehat{\theta}_t) - \sum_{t=1}^{T} \ell_t(\theta_t).$$

Our goal is to develop an online convex optimization algorithm with low (sublinear in T) regret relative to a broad family of comparator sequences. Previous work proposed algorithms which yielded regret of $O(\sqrt{T})$ for the relatively small family of *static* comparators, where $\theta_t = \theta$ for some $\theta \in \Theta$ and all t. In contrast, our main result is an algorithm which incorporates a dynamical model, denoted $\Phi_t : \Theta \mapsto \Theta$, and admits a tracking regret bound of the form $O(\sqrt{T}[1 + \sum_{t=1}^{T-1} \|\theta_{t+1} - \Phi_t(\theta_t)\|])$.

1.3 Online convex optimization preliminaries

One common approach to forming the predictions $\widehat{\theta}_t$, Mirror Descent (MD) [40, 8], consists of solving the following optimization problem:

$$\widehat{\theta}_{t+1} = \underset{\theta \in \Theta}{\arg\min}\, \eta_t \langle \nabla \ell_t(\widehat{\theta}_t), \theta \rangle + D(\theta \| \widehat{\theta}_t), \tag{1.1}$$

where $\nabla \ell_t(\theta)$ denotes an arbitrary subgradient of ℓ_t at θ, $D(\theta \| \widehat{\theta}_t)$ is the *Bregman divergence* [12, 15] between θ and $\widehat{\theta}$, and $\eta_t > 0$ is a step size parameter. Let $\psi : \Theta \to \mathbb{R}$ denote a continuously differentiable function that is σ-strongly convex for some parameter $\sigma > 0$ and some norm $\|\cdot\|$:

$$\psi(\theta_1) \geq \psi(\theta_2) + \langle \nabla \psi(\theta_2), \theta_1 - \theta_2 \rangle + \frac{\sigma}{2} \|\theta_1 - \theta_2\|^2 \tag{1.2}$$

The Bregman divergence associated with ψ is

$$\begin{aligned} D(\theta_1\|\theta_2) &\triangleq \psi(\theta_1) - \psi(\theta_2) - \langle \nabla\psi(\theta_2), \theta_1 - \theta_2\rangle \\ &\geq \frac{\sigma}{2}\|\theta_1 - \theta_2\|^2 \end{aligned} \tag{1.3}$$

An important consequence of this definition is the following generalization of the law of cosines: for all $\theta_1, \theta_2, \theta_3 \in \Theta$

$$\begin{aligned} D(\theta_1\|\theta_2) =& D(\theta_3\|\theta_2) + D(\theta_1\|\theta_3) \\ &+ \langle \nabla\psi(\theta_2) - \nabla\psi(\theta_3), \theta_3 - \theta_1\rangle. \end{aligned} \tag{1.4}$$

The MD approach is a generalization of online learning algorithms such as online gradient descent [60] and weighted majority [33]. Several recently proposed methods consider the data-fit term separately from the regularization term [22, 57, 31]. For instance, consider Composite Objective Mirror Descent (COMID) [22], where:

$$\widehat{\theta}_{t+1} = \underset{\theta\in\Theta}{\arg\min}\, \eta_t\langle\nabla f_t(\widehat{\theta}_t), \theta\rangle + \eta_t r(\theta) + D(\theta\|\widehat{\theta}_t). \tag{1.5}$$

This formulation is helpful when the regularization function $r(\theta)$ promotes sparsity in θ, and helps ensure that the individual $\widehat{\theta}_t$ are indeed sparse, rather than approximately sparse as are the solutions to the MD formulation.

1.3.1 Static regret

In much of the online learning literature, the comparator sequence is constrained to be static or time-invariant. In this paper we refer to the regret with respect to a static comparator as *static regret*:

Definition 2 (Static regret). *The* static regret *of* $\widehat{\boldsymbol{\theta}}_T$ *is*

$$R_T(\boldsymbol{\theta}_T) \triangleq \sum_{t=1}^{T} \ell_t(\widehat{\theta}_t) - \min_{\theta\in\Theta} \sum_{t=1}^{T} \ell_t(\theta).$$

Static regret bounds are useful in characterizing how well an online algorithm performs relative to, say, a loss-minimizing batch algorithm with access to all the data simultaneously. More generally, static regret bounds compare the performance of the algorithm against a static point which can be chosen with full knowledge of the data.

1.3.2 Tracking regret

Static regret fails to illuminate the performance of online algorithms in dynamic settings where the underlying parameters may be changing in time. Performance relative to a temporally-varying or dynamic comparator sequence has been studied previously in the literature in the context of tracking regret (also known as shifting regret) [28, 16], and the closely-related concept of adaptive regret [33, 26].

In particular, tracking regret compares the output of the online algorithm to a sequence of points $\theta_1, \theta_2, \ldots, \theta_T$ which can be chosen collectively with full knowledge of the data. This is a fair comparison for a batch algorithm that detects and fits to drift in the data, instead of fitting a single point. Frequently, in order to bound tracking regret there needs to be a measure of the *complexity* of the sequence $\theta_1, \theta_2, \ldots, \theta_T$, characterized via a measure of the temporal variability of the sequence, such as

$$V(\boldsymbol{\theta}_T) \triangleq \sum_{t=1}^{T-1} \|\theta_{t+1} - \theta_t\|.$$

If this complexity is allowed to be very high, we could imagine that the comparator series would fit the datastream and associated series of losses closely and hence generalize poorly. Conversely, if this complexity is restricted to be 0, the tracking regret is equivalent to static regret. Generally, sublinear tracking regret is only possible when the comparator sequence $\boldsymbol{\theta}_T$ is piecewise constant (where $\theta_{t+1} - \theta_t = 0$ for all but a few t) or varying quite slowly over time – that is, for a small family of comparators.

Summary of the most important results

2.1 Dynamical models in online convex programming

In contrast to previous tracking regret bounds, we develop methods and tracking regret bounds which scale with $\sum_t^{T-1} \|\theta_{t+1} - \Phi_t(\theta_t)\|$, where $\{\Phi_t\}, t = 1, 2, \ldots$ is a sequence of dynamical models, yielding small regret bounds for much broader classes of dynamic comparator sequences. Specifically, we propose the alternative to (1.1) and (1.5) in Algorithm 1, which we call *Dynamic Mirror Descent (DMD)*. By including Φ_t in the process, we effectively search for a predictor which (a) attempts to minimize the loss and (b) which adheres to the dynamical model Φ_t. This is similar to a stochastic filter which alternates between using a dynamical model to update the "state", and then uses this state to perform the filtering action. A key distinction of our approach and analysis, however, is that we make no assumptions about Φ_t's relationship to the observed data. Our approach effectively includes dynamics into the COMID approach.[1] Indeed, for a case with no dynamics, so that $\Phi_t(\theta) \equiv \theta$ for all θ and t, our method is equivalent to COMID.

Algorithm 1 Dynamic mirror descent (DMD) with known dynamics

```
Given decreasing sequence of step sizes $\eta_t > 0$
Initialize $\widehat{\theta}_1 \in \Theta$.
for $t = 1, \ldots, T$ do
    Observe $x_t$ and incur loss $\ell_t(\widehat{\theta}_t)$
    Receive dynamical model $\Phi_t$
    Set
        $\widetilde{\theta}_{t+1} = \arg\min_{\theta \in \Theta} \eta_t \langle \nabla f_t(\widehat{\theta}_t), \theta \rangle + \eta_t r(\theta) + D(\theta \| \widehat{\theta}_t)$    (2.1a)
        $\widehat{\theta}_{t+1} = \Phi_t(\widetilde{\theta}_{t+1})$    (2.1b)
end for
```

Our main result uses the following assumptions:

- For all $t = 1, \ldots, T$ the functions ℓ_t and ψ are Lipschitz with constants G and M respectively, such that $\|\nabla \ell_t(\theta)\|_* \leq G$ and $\|\psi(\theta)\|_* \leq M$ for all $\theta \in \Theta$. The function $\| \cdot \|_*$ used in these assumptions is the dual to the norm in (1.2).
- There exists a constant $D_{\max}$ such that $D(\theta_1 \| \theta_2) \leq D_{\max}$ for all $\theta_1, \theta_2 \in \Theta$.
- For all $t = 1, \ldots, T$, the transformation Φ_t has a maximum distortion factor Δ_Φ such that $D(\Phi_t(\theta_1) \| \Phi_t(\theta_2)) - D(\theta_1 \| \theta_2) \leq \Delta_\Phi$ for all $\theta_1, \theta_2 \in \Theta$. When $\Delta_\Phi \leq 0$ for all t, we say that Φ_t satisfies the contractive property.

[1] Rather than considering COMID, we might have used other online optimization algorithms, such as the Regularized Dual Averaging (RDA) method [57], which has been shown to achieve similar performance with more regularized solutions. However, to the best of our knowledge, no tracking or shifting regret bounds have been derived for dual averaging methods (regularized or otherwise). Recent results on the equivalence of COMID and RDA [37] suggest that the bounds derived here might also hold for a variant of RDA, but proving this remains an open problem.

2.1.1 Tracking regret bound

Theorem 3 (Tracking regret of dynamic mirror descent)**.** *Let Φ_t be a dynamical model such that $\Delta_\Phi \leq 0$ for $t = 1, 2, \ldots, T$ with respect to the Bregman used in 2.1. Let the sequence $\widehat{\boldsymbol{\theta}}_T$ be generated using Alg. 1 using a non-increasing series $\eta_{t+1} \leq \eta_t$, with a convex, Lipschitz function ℓ_t on a closed, convex set Θ, and let $\boldsymbol{\theta}_T$ be an arbitrary sequence in Θ^T. Then*

$$R_T(\boldsymbol{\theta}_T) \leq \frac{D_{\max}}{\eta_{T+1}} + \frac{2M}{\eta_T} V_\Phi(\boldsymbol{\theta}_T) + \frac{G^2}{2\sigma} \sum_{t=1}^{T} \eta_t$$

$$\text{with } V_\Phi(\boldsymbol{\theta}_T) \triangleq \sum_{t=1}^{T-1} \|\theta_{t+1} - \Phi_t(\theta_t)\|$$

where $V_\Phi(\boldsymbol{\theta}_T)$ measures variations or deviations of the comparator sequence $\boldsymbol{\theta}_T$ from the sequence of dynamical models $\Phi_1, \Phi_2, \ldots, \Phi_T$. If $\eta_t \propto \frac{1}{\sqrt{t}}$, then for some $C > 0$ independent of T,

$$R(\boldsymbol{\theta}_T) \leq C\sqrt{T}\,(1 + V_\Phi(\boldsymbol{\theta}_T))$$

This bound scales with the comparator sequence's deviation from the sequence of dynamical models $\{\Phi_t\}_{t>0}$ – a stark contrast to previous tracking regret bounds which are only sublinear for comparators which change slowly with time or at a small number of distinct time instances. Note that when Φ_t corresponds to an identity operator, the bound in Theorem 3 corresponds to existing tracking or shifting regret bounds [17, 16].

It is intuitively satisfying that this measure of variation, $V_\Phi(\boldsymbol{\theta}_T)$, appears in the tracking regret bound. First, if the comparator actually follows the dynamics, this variation term will be very small, leading to low tracking regret. This fact holds whether Φ_t is part of the generative model for the observations or not. Second, we can get a dynamic analog of static regret, where we enforce $V_\Phi(\boldsymbol{\theta}_T) = 0$. This is equivalent to saying that the batch comparator is fitting the best single trajectory using Φ_t instead of the best single point. Using this, we would recover a bound analogous to a static regret bound in a stationary setting.

The condition that $\Delta_\Phi \leq 0$ is similar to requiring that Φ_t be a contractive mapping. This restriction is important; without it, any poor prediction made at one time step could be exacerbated by repeated application of the dynamics. For instance, linear dynamic models with all eigenvalues less than or equal to unity satisfy this condition with respect to the squared ℓ_2 Bregman Divergence, similar in spirit to restrictions made in more classical adaptive filtering work such as [25].

Notice also that if $\Phi_t(\theta) = \theta$ in for all t, then Theorem 3 gives a novel, previously unknown tracking regret bound for COMID.

2.1.2 Data-dependent dynamics

An interesting example of dynamical models is the class of data-dependent dynamical models. In this regime the state of the system at a given time is not only a function of the previous state, but also the actual observations. One key example of this scenario arises in self-exciting point processes, where the state of the system is directly related to the previous, stochastic observations. Our algorithm can account for such models since the function $\Phi_t(\theta)$ is time varying, and therefore can implicitly depend on all data up to time t, i.e. $\Phi_t(\theta) = \Phi_t(\theta, x_1, x_2, \ldots, x_t)$. Our regret bounds therefore scale with how well the comparator series matches these data dependent dynamics:

$$R(\boldsymbol{\theta}_T) \leq C\left(\sqrt{T}\left[1 + \sum_{t=1}^{T-1} \|\theta_{t+1} - \Phi_t(\theta_t, x_1, \dots, x_t)\|\right]\right).$$

Notice now that the data plays a part in the regret bounds, whereas before we only measured the variation of the comparator. Data-dependent regret bounds are not new. Concurrent related work considers online algorithms where the data sequence is described by a "predictable process" [43]. The basic idea of that paper is that if one has a sequence functions M_t which predict x_t based on $x_1, x_2, \dots, x_{t-1}$, then the output of a standard online optimization routine should be combined with the predictor generated by M_t to yield tighter regret bounds that scale with $(\sum_t \|x_t - M_t(x_1, \dots, x_{t-1})\|^2)^{1/2}$. However, [43] only works with static regret (*i.e.,* regret with respect to a static comparator) and their regret has a variation term that expresses the deviation of the *input data* from the underlying process. In contrast, our tracking regret bounds scale with the deviation of a *comparator sequence* from a prediction model.

2.2 Prediction with a finite family of dynamical models

DMD in the previous section uses a single sequence of dynamical models. In practice, however, we may not know the best dynamical model to use, or the best model may change over time in nonstationary environments. To address this challenge, we assume a finite set of candidate dynamical models $\{\Phi_{1,t}, \Phi_{2,t}, \dots, \Phi_{N,t}\}$ at every time t, and describe a procedure which uses this collection to adapt to nonstationarities in the environment. In particular we establish tracking regret bounds which scale not with the deviation of a comparator from a single dynamical model, but with how it deviates from a *series of different dynamical models on different time intervals* with at most m switches. These switches define $m+1$ different time segments $[t_i, t_{i+1}-1]$ with time points $1 = t_1 < \cdots < t_{m+2} = T$. We can bound the regret associated with the best dynamical model on each time segment and then bound the overall regret using a Prediction with Experts Advice algorithm

Our *dynamic fixed share* (DFS) estimate is presented in Algorithm 2. Let $\widehat{\theta}_{i,t}$ denote the output of Alg. 1 using dynamical models $\Phi_{i,1}, \Phi_{i,2}, \dots, \Phi_{i,t}$; we choose $\widehat{\theta}_t$ by using the Fixed Share forecaster on these outputs.[2] In Fixed Share, each expert (here, each sequence of candidate dynamical models) is assigned a weight that is inversely proportional to its cumulative loss at that point yet with some weight shared amongst all the experts, so that an expert with very small weight can quickly regain weight to become the leader [27, 16].

In this update, $\lambda \in (0,1)$ is a parameter which controls how much of the weight is shared amongst the experts. By sharing some weight, it allows experts with high loss, and therefore low

[2] There are many algorithms from the Prediction with Expert Advice literature which can be used to form a single prediction from the predictions created by the set of dynamical models. We use the Fixed Share algorithm [27] as a means to combine estimates with different dynamics; however, other methods could be used with various tradeoffs. One of the primary drawbacks of the Fixed Share algorithm is that an upper bound on the number of switches m must be known a priori. However, this method has a simple implementation and tracking regret bounds. One common alternative to Fixed Share allows the switching parameter (λ in Alg. 2) to decrease to zero as the algorithm runs [30, 1]. This has the benefit of not requiring knowledge about the number of switches, but comes at the price of higher regret. Alternative expert advice algorithms exist which decrease the regret but increase the computational complexity. For a thorough treatment of existing methods see [24].

Algorithm 2 Dynamic fixed share (DFS)

```
Given decreasing sequence of step sizes $\eta_t > 0$ and $\eta_r > 0$
Initialize $\widehat{\theta}_1 \in \Theta$, $\widehat{\theta}_{i,1} \in \Theta$ and $w_{i,1} = \frac{1}{N}$ for $i = 1, \dots, N$, $\lambda \in (0,1)$, and $\eta_t, \eta_r > 0$.
for $t = 1, \dots, T$ do
  Observe $x_t$ and incur loss $\ell_t(\widehat{\theta}_t)$
  Receive dynamical model $\Phi_{i,t}$ for $i = 1, \dots, N$
  for $i = 1, \dots, N$ do
    Set
    $$\begin{aligned} \widetilde{w}_{i,t+1} &= \frac{w_{i,t} \exp\left(-\eta_r \ell_t\left(\widehat{\theta}_{i,t}\right)\right)}{\sum_{j=1}^{N} w_{j,t} \exp\left(-\eta_r \ell_t\left(\widehat{\theta}_{j,t}\right)\right)} \\ w_{i,t+1} &= \frac{\lambda}{N} + (1-\lambda)\widetilde{w}_{i,t} \\ \widetilde{\theta}_{i,t+1} &= \arg\min_{\theta \in \Theta} \eta_t \langle \nabla f_t(\widehat{\theta}_{i,t}), \theta \rangle + \eta_t r(\theta) + D(\theta \| \widehat{\theta}_{i,t}) \\ \widehat{\theta}_{i,t+1} &= \Phi_{i,t}(\widetilde{\theta}_{i,t+1}) \end{aligned}$$
  end for
  Set
  $$\widehat{\theta}_{t+1} = \sum_{i=1}^{N} w_{i,t+1} \widehat{\theta}_{i,t+1}$$
end for
```

weight, to quickly regain weight if they start performing well. This is the mechanism that allows fast switching between experts.

Theorem 4 (Tracking regret of DFS algorithm)**.** *Assume all the candidate dynamic sequences are contractive such that $\Delta_\Phi \leq 0$ for $\Phi_{i,t}$ for all $t = 1, .., T$ and $i = 1, ..., N$ with respect to the Bregman Divergence in alg 1. Then for some $C > 0$, the dynamic fixed share algorithm in Algorithm 2 with parameter λ set equal to $\frac{m}{T-1}$, $\eta_r = \sqrt{\frac{8((m+1)\log(N)+m\log(T)+1)}{T}}$ and $\eta_t = 1/\sqrt{t}$ with a convex, Lipschitz function ℓ_t on a closed, convex set Θ, has tracking regret*

$$\begin{aligned} R_T(\boldsymbol{\theta}_T) &= \sum_{t=1}^{T} \ell_t(\widehat{\theta}_t) - \sum_{t=1}^{T} \ell_t(\theta_t) \\ &\leq C\left(\sqrt{T}\left(\sqrt{(m+1)\log N + m\log T} + V^{(m+1)}(\boldsymbol{\theta}_T)\right)\right), \end{aligned}$$

where

$$V^{(m+1)}(\boldsymbol{\theta}_T) \triangleq \min_{t_2,\ldots,t_{m+1}} \sum_{k=1}^{m+1} \min_{i_k \in \{1,\ldots,N\}} \sum_{t=t_k}^{t_{k+1}-1} \|\theta_{t+1} - \Phi_{i_k,t}(\theta_t)\|$$

measures the deviation of the sequence $\boldsymbol{\theta}_T$ from the best sequence of dynamical models with at most m switches (where m does not depend on T).

Note that the family of comparator sequences $\boldsymbol{\theta}_T$ for which $R_T(\boldsymbol{\theta}_T)$ scales sublinearly in T is *significantly* larger than the set of comparators yielding sublinear regret for MD.

If T is not known in advance the doubling trick [17] can be used, where temporary time horizons are set of increasing length. Note that $V^{(m+1)}(\boldsymbol{\theta}_T) \leq V_{\Phi_{i,t}}(\boldsymbol{\theta}_T)$ for any fixed $i \in \{1, \ldots, N\}$, thus this approach yields a lower variation term than using a fixed dynamical model. However, we incur some loss by not knowing the optimal number of switches m or when the optimal switching times are.

2.3 Parametric dynamical models

Rather than having a finite family of dynamical models, as we did in Section 2.2, we may consider a parametric family of dynamical models, where the parameter $\alpha \in \mathbb{R}^n$ of Φ_t is allowed to vary across a closed, convex domain, denoted $\mathcal{A}$. In other words, we consider $\Phi_t : \Theta \times \mathcal{A} \mapsto \Theta$. In this context we would like to *jointly* predict both α and θ. One might consider defining $\zeta \triangleq (\theta; \alpha)$ as the concatenation of θ and α, and then generating a sequence of $\widehat{\zeta}_t$'s using COMID (1.5) or DMD (2.1). However, the COMID regret would not capture deviations of a comparator sequence of predictions from a series of dynamical models as desired. To use DMD, we would need to define a dynamical model $\Psi : \Theta \times \mathcal{A} \mapsto \Theta \times \mathcal{A}$, so that $\Psi(\theta, \alpha) = (\Phi(\theta, \alpha), \alpha)$, and use this in place of Φ_t in (2.1). However, Ψ is not contractive for most Φ of interest, so the DMD regret bounds would not hold.

To address these challenges, we consider two approaches. First, in Section 2.3.1 we consider tracking only a finite subset of the possible model parameters, in a manner similar to when we had a finite collection of possible dynamical models, which provide a "covering" of the parameter space. In this case, the overall regret and computational complexity both depend on the resolution of the covering set. Second, in Section 2.3.2, we consider a special family of additive dynamical models; in this setting, we can efficiently learn the optimal dynamics.

2.3.1 Covering the set of dynamical models

In this section we show that by tracking a subset which appropriately covers the entire space of candidate models, we can bound the overall regret, as well as bound the number of parameter values we have to track, and the inherent tradeoff between the two. We propose to choose a finite collection of parameters from a closed, convex set $\mathcal{A}$ and perform DFS (Alg. 2) on this collection. We specifically consider the case where the true dynamical model $\alpha^* \in \mathcal{A}$ is unchanging in time and use DFS with $m = 0$. (Fixed share with $m = 0$ amounts to the Exponentially Weighted Averaging Forecaster [55, 33, 17].) In the below, for any $\alpha \in \mathcal{A}$, let

$$V_\Phi(\boldsymbol{\theta}_T, \alpha) \triangleq \sum_{t=1}^{T-1} \|\theta_{t+1} - \Phi_t(\theta_t, \alpha)\|.$$

Theorem 5 (Covering sets of dynamics parameter space)**.** *Let $\varepsilon_N > 0$ and $\mathcal{A}_N$ denote a covering set for $\mathcal{A}$ with cardinality N, such that for every $\alpha \in \mathcal{A}$, there is some $\alpha' \in \mathcal{A}_N$ such that $\|\alpha - \alpha'\| \leq \varepsilon_N$. Define candidate dynamical models as $\Phi_t(\cdot, \alpha)$ for $\alpha \in \mathcal{A}_N$ and assume they are all contractive with respect to the Bregman Divergence used in Alg. 1. If $\|\Phi_t(\theta, \alpha) - \Phi_t(\theta, \beta)\| \leq L\|\alpha - \beta\|$ for some $L > 0$ for all $\alpha, \beta \in \mathcal{A}$, then for some constant $C > 0$, the DFS algorithm with $\eta_t = \frac{1}{\sqrt{t}}, \eta_r = \sqrt{\frac{2\log(N)}{T}}, \lambda = 0$ yields a tracking regret bounded by*

$$C\left(\sqrt{T}\left[\sqrt{\log(N)} + \min_{\alpha \in \mathcal{A}} V_\Phi(\boldsymbol{\theta}_T, \alpha) + T\varepsilon_N\right]\right).$$

Intuitively, we know that if we set ε_N to be very small we will have good performance because any possible parameter value $\alpha \in \mathcal{A}$ would have to be close to a candidate dynamic; however, we would need to choose many candidates. Conversely, if we run DFS on only a few candidate models, it will be computationally much more efficient but our total regret will grow due to parameter mismatch.

Corollary 6. *Assume $\mathcal{A} \subseteq [A_{\min}, A_{\max}]^n$, and let $\gamma > 0$ be given. Let $k = \lceil (A_{\max} - A_{\min}) n T^\gamma / 2 \rceil$ and $\partial = (A_{\max} - A_{\min})/(2k)$; let $\mathcal{A}_N = \{A_{\min} + \partial, A_{\min} + 3\partial, \ldots, A_{\min}(2k-1)\partial\}^n$ correspond to an n-dimensional grid with k^n grid points over $\mathcal{A}$. Then*

$$\max_{\alpha \in \mathcal{A}} \min_{\alpha' \in \mathcal{A}_N} \|\alpha - \alpha'\|_1 \leq T^{-\gamma}.$$

Additionally, the total number of grid points is upper bounded by

$$N \leq \left(\frac{(A_{\max} - A_{\min}) n T^\gamma}{2} + 1\right)^n = O(T^{\gamma n})$$

*Under the assumptions of Theorem 5, with this set $\mathcal{A}_N$ and using the fact that norms are equivalent on finite-dimensional vectors (*i.e., *there's a finite $Z > 0$ such that $\|\alpha - \beta\|_1 \leq Z\|\alpha - \beta\|$ for any $\alpha, \beta \in \mathcal{A}$ for any norm), we get the following bound on regret for some constant $C > 0$.*

$$R_T(\boldsymbol{\theta}_T) \leq C\left(\sqrt{T}\left[\sqrt{\gamma n \log(T)} + \min_{\alpha \in A} V_\Phi(\boldsymbol{\theta}_T, \alpha)\right] + T^{1-\gamma n}\right)$$

Here we have an explicit tradeoff between regret and computationally accuracy controlled by γ, since the computational computational complexity is linear in $N = O(T^{\gamma n})$. We can further control the tradeoff between computation complexity and performance by allowing ε_N to vary in time. This could be done by using the doubling trick, setting temporary time horizons, and then refining the grid once the temporary time horizon is reached using a slightly different experts algorithm which could account for the changing number of experts as in [47].

2.3.2 Additive dynamics in exponential families

The approach described above for generating a covering set of dynamical models may be effective when the dimension of parameters is small; however, in higher dimensions, this approach can require significant computational resources. In this section, we consider an alternative approach that only requires the computation of predictions for a single dynamical model. We will see that in some settings, the prediction produced by Dynamic Mirror Descent (DMD) and a certain set of parameters for the dynamic model can quickly be converted to the prediction for a different set of parameters. While the method described in this section is efficient and admits strong regrets bounds, it is applicable only for loss functions derived from exponential families.

The basics of exponential families are described in [2, 56], and mirror descent in this setting is explored in [5, 42]. We assume some $\phi : \mathsf{X} \to \mathbb{R}^d$ which is a measurable function of the data, and let ϕ_k, $k = 1, 2, \ldots, d$, denote its components:

$$\phi(x) = (\phi_1(x), ..., \phi_d(x))^T .$$

We use the specific loss function

$$\ell_t(\theta) = -\log p_\theta(x_t) \tag{2.2a}$$

where

$$p_\theta(x) \triangleq \exp\{\langle \theta, \phi(x) \rangle - Z(\theta)\} \tag{2.2b}$$

for a sufficient statistic ϕ and $Z(\theta) \triangleq \log \int \exp\{\langle \theta, \phi(x) \rangle\} dx$, known as the *log-partition function*, ensures that $p_\theta(x)$ integrates to a constant independent of θ. Furthermore, as in [5, 42], we use the Bregman divergence corresponding to the Kullback-Liebler divergence between two members of the exponential family:

$$D(\theta_1 \| \theta_2) = Z(\theta_1) - Z(\theta_2) - \langle \nabla Z(\theta_2), \theta_1 - \theta_2 \rangle.$$

In our analysis we will be using the *Legendre–Fenchel dual* of Z [29, 11]:

$$Z^*(\mu) \triangleq \sup_{\theta \in \Theta} \{ \langle \mu, \theta \rangle - Z(\theta) \} .$$

Let Θ^* denote the image of Int Θ under the gradient mapping ∇Z i.e. $\Theta^* = \nabla Z(\text{Int } \Theta)$. An important fact is that the gradient mappings ∇Z and ∇Z^* are inverses of one another [8, 17, 39]:

$$\left.\begin{array}{r} \nabla Z^*(\nabla Z(\theta)) = \theta \\ \nabla Z(\nabla Z^*(\mu)) = \mu \end{array}\right\} \qquad \forall \theta \in \text{Int } \Theta, \mu \in \text{Int } \Theta^*$$

Following [17], we may refer to the points in Int Θ as the *primal points* and to their images under ∇Z as the *dual points*. For simplicity of notation, in the sequel we will write $\mu = \nabla Z(\theta)$, $\theta = \nabla Z^*(\mu)$, $\widehat{\mu}_t = \nabla Z(\widehat{\theta}_t)$, etc.

Additionally, we will use a dynamical model that takes on a specific form:

$$\Phi_t(\theta, \alpha) = \nabla Z^*(A_t \nabla Z(\theta) + B_t \alpha + c_t) \tag{2.3}$$

for $\theta \in \text{Int } \Theta, c_t \in \mathbb{R}^d$, $\alpha \in \mathbb{R}^n$, $A_t \in \mathbb{R}^{d \times d}$, and $B_t \in \mathbb{R}^{d \times n}$. A_t, B_t and c_t are considered known. Using these dynamics, we let $\widehat{\theta}_{\alpha,t}$ denote the output of DMD (Alg. 1) at time t and $\widehat{\mu}_{\alpha,t}$ be its dual. Under all these conditions, we have the following Lemma.

Lemma 7. *For any $\alpha, \beta \in \mathcal{A}$, let $\widehat{\mu}_{\alpha,1} = \widehat{\mu}_{\beta,1}$ be the duals of the initial prediction for DMD and $K_1 = \mathbf{0} \in \mathbb{R}^{d \times n}$. Additionally assume that the minimizer of equation 2.1a is a point in Int Θ for any parameter $\alpha \in \mathcal{A}$. Then the DMD prediction under a dynamical model parameterized by α can be calculated directly from the DMD prediction under a dynamical model parameterized by β for $t > 0$ as*

$$\widehat{\mu}_{\alpha,t} = \widehat{\mu}_{\beta,t} + K_t(\alpha - \beta)$$

where

$$K_t = (1 - \eta_{t-1})A_{t-1}K_{t-1} + B_{t-1}.$$

From Lemma 7, we see that the prediction for dynamical model α can be computed simply from the prediction using parameters β and the value K_t. This is a significant computational gain compared to DFS, where we had to keep track of predictions for each candidate dynamical model individually and therefore needed to bound the number of experts for tractability.

Algorithm 3 leverages Lemma 7 to simultaneously track both $\widehat{\theta}_t$ and the best dynamical model parameter α. In this algorithm, $\widetilde{\ell}_t$ is the function defined as

$$\widetilde{\ell}_t(\mu) \triangleq \ell_t(\nabla Z^*(\mu)) \equiv \ell_t(\theta).$$

The basic idea is the following: we use mirror descent to compute an estimate of the best dynamical model parameter, compute the DMD prediction associated with that parameter, and then use DMD to update that prediction for the next round.

Algorithm 3 Dynamic mirror descent (DMD) with parametric additive dynamics

```
Given decreasing sequence of step sizes $\rho_t, \eta_t > 0$
Initialize $\widehat{\alpha}_1 = \mathbf{0}$, $K_1 = \mathbf{0}$, $\widehat{\theta}_1 \in \Theta$, $\widehat{\mu}_1 = \nabla Z(\widehat{\theta}_1)$
for $t = 1, ..., T$ do
    Observe $x_t$
    Incur loss $\ell_t(\widehat{\theta}_t) = -\langle \widehat{\theta}_t, \phi(x_t) \rangle + Z(\widehat{\theta}_t)$
    Set $g_t(\alpha) = \widetilde{\ell}_t(\widehat{\mu}_{\alpha,t}) \equiv \ell_t(\widehat{\theta}_{\alpha,t})$
    Set $\widehat{\alpha}_{t+1} = \mathrm{proj}_{\mathcal{A}}\left(\widehat{\alpha}_t - \rho_t \nabla g_t(\widehat{\alpha}_t)\right)$
    Set $\mu'_{t+1} = \widehat{\mu}_t + K_t(\widehat{\alpha}_{t+1} - \widehat{\alpha}_t)$
    Set $\widetilde{\mu}_{t+1} = (1 - \eta_t)\mu'_{t+1} + \eta_t \phi(x_t)$
    Set $\widetilde{\theta}_{t+1} = \nabla Z^*(\widetilde{\mu}_{t+1})$
    Set $\widehat{\theta}_{t+1} = \Phi_t(\widetilde{\theta}_{t+1}, \widehat{\alpha}_{t+1})$
    Set $K_{t+1} = (1 - \eta_t)A_t K_t + B_t$
end for
```

Theorem 8. *Assume that the observation space X is bounded. Let $\Theta \subset \mathbb{R}^d$ be a bounded, convex set satisfying the following properties for a given constant $H > 0$:*

- *For all $\theta \in \Theta$,*

$$Z(\theta) \triangleq \int_{\mathsf{X}} \exp\left\{\langle \theta, \phi(x) \rangle\right\} d\nu(x) < +\infty.$$

- *For all $\theta \in \Theta$, $\nabla^2 Z(\theta) \succeq 2H I_{d \times d}$.*

- *Let f_t denote the objective function in (2.1a). For every $x \in \mathsf{X}$ and $t \in \{1, 2, 3, \ldots\}$, the solution to $\underset{\theta \in \Theta}{\arg\min}\, f_t(\theta)$ occurs where $\nabla f_t = \mathbf{0}$.*

If the assumptions of Lemma 7 hold, and $\Phi_t(\theta, \alpha)$ is contractive for all $\alpha \in \mathcal{A}$ with respect to the Bregman Divergence induced by $Z(\theta)$, and loss function is of the form (2.2) and

$$\widetilde{\ell}_t(\mu) \triangleq \ell_t(\nabla Z^*(\mu))$$

is convex in μ, and $\eta_t, \rho_t \propto 1/\sqrt{t}$, then the tracking regret associated with Algorithm 3 for dynamical models of the form (2.3) is

$$R_T(\boldsymbol{\theta}_T) \leq C\sqrt{T}\left(1 + \min_{\alpha \in \mathcal{A}} V_\Phi(\boldsymbol{\theta}_T, \alpha)\right)$$

for some constant $C > 0$.

Theorem 8 shows that Algorithm 3 allows us to simultaneous track predictions and dynamics, and we perform nearly as well as if we knew the best dynamical models for the entire sequence in hindsight. While this approach is only applicable for specific forms of the loss functions and dynamical models, those forms arise in a wide variety of practical problems.

2.4 Experiments and results

As mentioned in the introduction, many online learning problems can benefit from the incorporation of dynamical models. In the below, we describe how the ideas described and analyzed in this paper might be applied to anomaly detection from streaming dynamic textures, compressive video reconstruction, and analysis of neuron firing rates within networks.

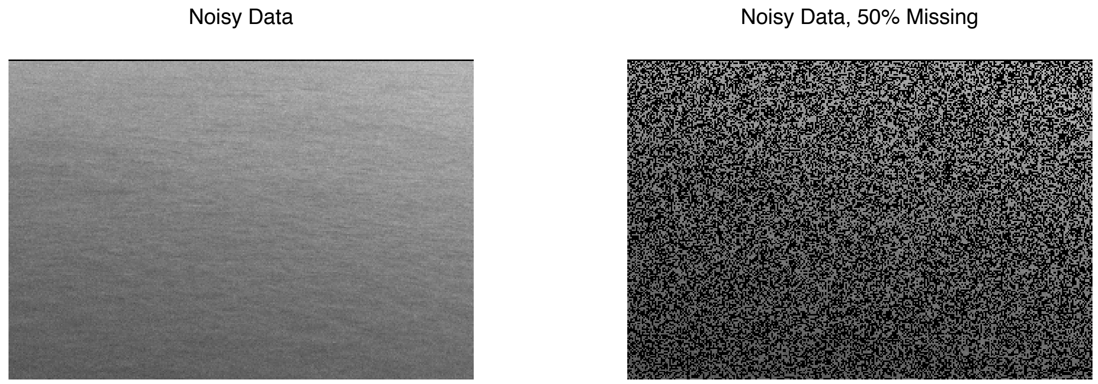
2.4.1 DMD experiment: dynamic textures with missing data

As mentioned in the introduction, sensors such as the Solar Data Observatory are generating data at unprecedented rates. Heliophysicists have physical models of solar dynamics, and often wish to identify portions of the incoming data which are inconsistent with their models. This "data thinning" process is an essential element of many big data analysis problems. We simulate an analogus situation in this section.

In particular, we consider a datastream corresponding to a dynamic texture [52][20] , where spatio-temporal dynamics within motion imagery are modeled using an autoregressive process. In this experiment, we consider a setting where "normal" autoregressive parameters are known, and we use these within DMD to track a scene from noisy image sequences with missing elements. (Missing elements arise in our motivating solar astronomy application, for instance, when cosmic rays interfere with the imaging detector array.) As suggested by our theory, the tracking will fail and generate very large losses when the posited dynamical model is inaccurate.

More specifically, the idea of dynamic textures is that a low dimensional, auto-regressive model can be used to simulate a video which replicates a moving texture such as flowing water, swirling fog, solar plasma flows, or rising smoke. This process is modeled in the following way:

$$\begin{aligned} \theta_t &= A\theta_{t-1} + Bu_t \\ x_t &= C_0 + C\theta_t + Dv_t \\ v_t, u_t &\sim \mathcal{N}(0, I). \end{aligned}$$



(a) Noisy realization of a synthetic dynamic water texture

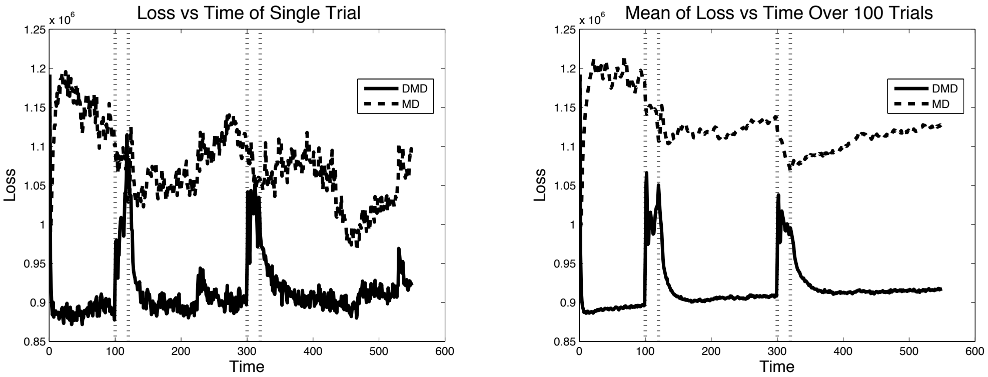
(b) Noisy realization of a synthetic dynamic water texture with 50% missing pixels

Figure 2.1: Dynamic textures simulation setup

In the above, θ_t denotes the true underlying parameters of the system, and x_t the observations. The matrix A is the autoregressive parameters of the system, which will be unique for the type of texture desired, C_0 the average background intensity, C is the sensing matrix which is usually a tall matrix, and B and D encode the strength of the driving and observation noises respectively. Using the toolbox developed in [44] and samples of a 220 by 320 pixel ocean scene [20], we learned two sets of parameters $A, A' \in \mathbb{R}^{50\times 50}$, one representing the water flowing when the data is played forward, and the other when played backwards, as well as corresponding parameters $C_0 \in \mathbb{R}^{70400}, C, C' \in \mathbb{R}^{70400\times 50}, B, B' \in \mathbb{R}^{50}$ and $D, D' \in \mathbb{R}^{70400}$. Parameters $\theta_t \ \in [-500, 500]^{50}$ and data $x_t \in [-500, 500]^{70400}$ were then generated using these parameters, with the parameters A', B', C', D' and C_0 on $t = 100, ..., 120$ and $t = 300, ..., 320$ and the parameters A, B, C, D, C_0 on the rest of $t = 1, ..., 550$ according to the above equations. Finally, every observation is corrupted by 50% missing values, chosen uniformly at random at every time point. Examples of the full noisy data, and data with missing values are shown in Figure 2.1.

The parameters A, C_0, and C were then used to define our (imperfect) dynamical model for DMD, $\Phi_t(\theta) = A\theta$, and a loss function $\ell_t(\theta) = \|P_t(C\theta - C_0 - x_t)\|_2^2$, where P_t is a linear operator accounting for the missing data. Note that B and D are not reflected in these choices despite playing a role in generating the data; our theoretical results hold regardless. We use $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ so the Bregman Divergence $D(x\|y) = \frac{1}{2}\|x - y\|_2^2$ is the usual squared Euclidean distance, and we perform no regularization ($r(\theta) = 0$). We set $\eta_t = \frac{1}{2\sqrt{t}}$, and ran 100 different trials comparing the DMD method to regular Mirror Descent (MD) to see the advantage of accounting for underlying dynamics. The results are shown in Figure 2.2.

There are a few important observations about this procedure. The first is that by incorporating the dynamic model, we produce an estimate which visually looks like the dynamic texture of interest, instead of the Mirror Descent prediction, which looks like a single snapshot of the water. Second, we can recover a good representation of the scene with a large amount of missing data, due to the autoregressive parameters being of a much lower dimension than the data itself. Finally, because we are using the dynamics of forward moving water, when the true data starts moving backward, *a change that is imperceptible visually*, the loss spikes, alerting us of the abnormal behavior.



(a) Loss curves for proposed dynamic mirror descent (DMD) method and mirror descent (MD) against time for a single trial.

(b) Loss curves for DMD and MD against time over 100 trials.

Figure 2.2: Simulation results for the experiment in Section 2.4.1. The vertical dashed lines indicate the intervals where the posited dynamical model was not reflected by the underlying data; note the sharp increases in the losses associated with DMD over those intervals, particularly in contrast with the losses associated with MD. Standard online learning methods like MD do not facilitate the detection of subsets of data which do not fit hypothesized physical models.

2.4.2 DFS experiment: social network analysis

To demonstrate the performance of Dynamic Mirror Descent (DMD) combined with the Fixed share algorithm (which we call *Dynamic Fixed Share (DFS)*), we consider two scenarios: reconstruction of a dynamic scene (*i.e.,* video) from sequential compressed sensing observations, and tracking connections in a dynamic social network.

Dynamical models have a rich history in the context of social network analysis [49], but we are unaware of their application in the context of online learning algorithms. To show how DMD can bridge this gap, we track the influence matrix of seats in the US Senate from 1795 to 2011 using roll call data (http://www.voteview.com/dwnl.htm). At time t, we observe the "yea" or "nay" vote of each Senator, which we represent with a $+1$ or -1. When a Senator's vote is unavailable (for instance, before a state joined the union), we use a 0. We form a length $p = 100$ vector of these votes indexed by the Senate seat, and denote this x_t.

Following [45], we form a loss function using a negative log Ising model *pseudolikelihood* to sidestep challenging issues associated with the partition function of the Ising model likelihood.

For a social network with p agents, $\theta_t \in [-1, 1]^{p \times p}$, where $(\theta_t)_{ab}$ corresponds to the correlation in voting patterns between agents a and b at time t. Let $\mathcal{V}$ denote the set of agents, $\mathcal{V} \backslash a$ the set of all agents except a, x_a the vote of agent a, and $\theta_a \triangleq \{\theta_{ab} : b \in \mathcal{V}\}$. Our loss function is

$$\begin{aligned}
\varphi_t^{(a)}(\theta_a) &\triangleq \log\left[\exp\left(2\theta_{aa}x_a + 2\textstyle\sum_{b \in \mathcal{V}\backslash a} \theta_{ab}x_a x_b\right) + 1\right] \\
f^{(a)}(\theta_a; x) &\triangleq -2\theta_{aa}x_a - 2\textstyle\sum_{b \in \mathcal{V}\backslash a} \theta_{ab}x_a x_b + \varphi_t^{(a)}(\theta_a) \\
f(\theta; x) &= \textstyle\sum_{a \in \mathcal{V}} f^{(a)}(\theta_a; x)
\end{aligned}$$

and $r(\theta) = \tau\|\theta\|_1$, where $\tau > 0$ is a tuning parameter; this loss is convex in θ. We let $\psi(\theta) = \frac{1}{2}\|\theta\|_2^2$

and use a dynamical model inspired by [49], where

$$(\Phi_i\theta)_{ab} = \begin{cases} (1-\alpha_i)\theta_{ab} + \alpha_i\theta_{ac^*}\theta_{bc^*} & \text{if } |\theta_{ac^*}\theta_{bc^*}| > |\widetilde{\theta}_{ab}| \\ \theta_{ab} & \text{otherwise} \end{cases},$$

where $c^* = \arg\min_c |\theta_{ac}\theta_{bc}|$.

The intuition is that if two members of the network share a strong common connection, they will become connected in time. We set $\alpha_i \in \{0, .001, .002, .003, .004\}$ for the different dynamical models. We set $\tau = .1$ and again set η using the doubling trick with time horizons at set at increasing powers of 10. As in [31], we find that regularizing (*e.g.,* thresholding) every 10 steps, instead of at each time step, allows for the values to grow above the threshold for meaningful relationships to be found.

The first plot in Figure 2.3 shows the average per round loss of each model, and the DFS estimator over a 30 year time window. We see that applying the dynamical model dramatically improves performance relative to COMD ($\alpha_i = 0$) and that the FS estimator aggregates the predictions successfully. The next plot shows the moving average losses for a few select Senators, where high loss corresponds to unpredictable behavior. Notice that John Kerry (D-MA) has generally very low loss, but spikes around 2006, but then drops again before a reelection campaign in 2008.

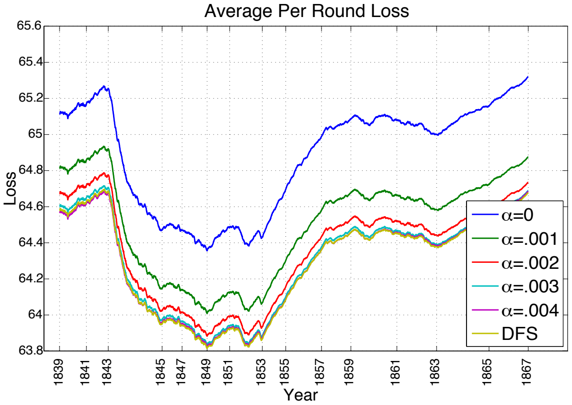


Figure 2.3: Tracking a dynamic social network. Losses for different dynamical models and the DFS predictions; $\alpha = 0$ corresponds to COMD.

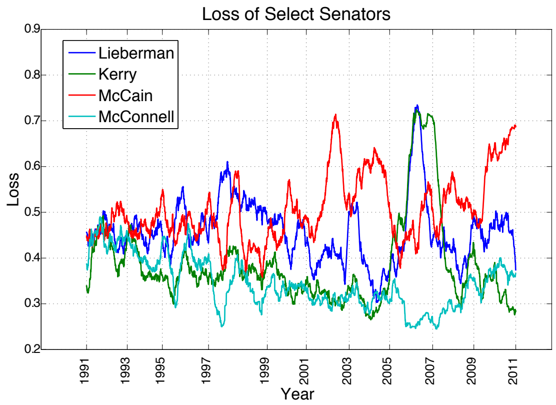


Figure 2.4: Predictability of individual senators. Low losses correspond to predictable, consistent voting behavior, while higher loss means less predictable

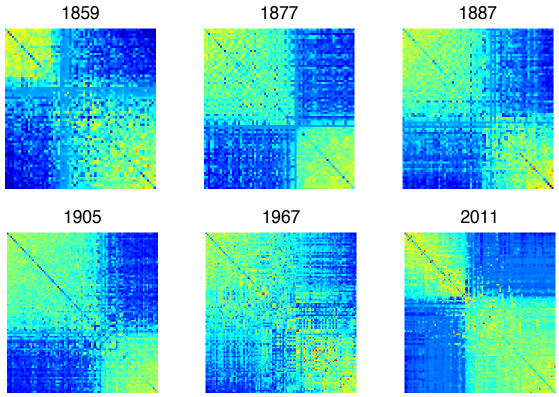


Figure 2.5: Influence matrices for select years spanning Civil War and Civil Rights Movement to present. We see tight factions forming in the mid- to late-1800s (post Civil War), followed by a time when the faction dissipate in the mid-1900s during the Civil Rights Movement and upheaval among southern Democrats. (Best viewed in color.)

By looking at the network estimates of the DFS estimator across time (as in Figure 2.5) we can see some interesting behavior in the network that corresponds to historical events, as described in the caption. Finally, we see factions again forming in more recent times. The seats are sorted separately for each matrix to emphasize groupings, which are strongly correlated with known political factions.

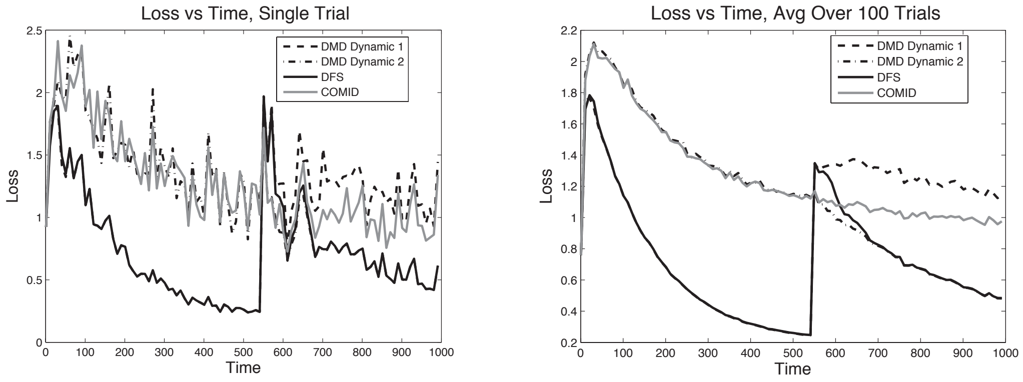
2.4.3 DFS experiment: compressive video reconstruction

There is increasing interest in using "big data" analysis techniques in applications like high-throughput microscopy, where scientists wish to image large collections of specimens. This work is facilitated by the development of novel microscopes, such as the recent fluorescence microscope based on structured illumination and compressed sensing principles [51]. However, measurements in such systems are acquired sequentially, posing significant challenges when imaging live specimens.

Knowledge of underlying motion in compressed sensing image sequences can allow for faster, more accurate reconstruction [34, 41, 46]. By accounting for the underlying motion in the image sequence, we can have an accurate prediction of the scene before receiving compressed measurements, and when the measurements are noisy and the number of observations is far less than the number of pixels of the scene, these predictions allow both fast and accurate reconstructions. If the dynamics are not accounted for, and previous observations are used as prior knowledge, the reconstruction could end up creating artifacts such as motion blur or overfitting to noise. There has been significant recent interest in using models of temporal structure to improve time series estimation from compressed sensing observations [4, 54]; the associated algorithms, however, are typically batch methods poorly suited to large quantities of streaming data. In this section we demonstrate that DMD helps bridge this gap.

In this section, we simulate fluorescence microscopy data generated by the system in [51] while imaging a paramecium moving in a 2-dimensional plane; the t^{th} frame is denoted θ_t (a 120×120 image stored as a length-14400 vector) which takes values between 0 and 1. The corresponding observation is $x_t = A_t\theta_t + n_t$, where A_t is a 50×14400 matrix with each element drawn iid from $\mathcal{N}(0,1)$ and n_t corresponds to measurement noise with $n_t \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 0.1$. This model coincides with several compressed sensing architectures [21, 51].

Our loss function uses $f_t(\theta) = \frac{1}{2\sigma^2 d}\|x_t - A_t\theta\|_2^2$ and $r(\theta) = \tau\|\theta\|_1$, where $\tau > 0$ is a tuning parameter. We construct a family of $N = 9$ dynamical models, where $\Phi_{i,t}(\theta)$ shifts the (unvectorized) frame, θ, one pixel in a direction corresponding to an angle of $2\pi i/(N-1)$ as well as a "dynamic" corresponding to no motion. (With the zero motion model, DMD reduces to COMID.) The true video sequence uses different dynamical models over $t = \{1, \ldots, 550\}$ (upward motion) and $t = \{551, \ldots, 1000\}$ (motion to the right). Finally, we use $\psi(\cdot) = \frac{1}{2}\|\cdot\|_2^2$ so the Bregman Divergence $D(x\|y) = \frac{1}{2}\|x-y\|_2^2$ is the usual squared Euclidean distance. The DMD sub-algorithms use $\eta_t = \frac{1}{\sqrt{t}}, \tau = .002$ and the DFS forecaster uses $\lambda = \frac{m}{T-1} = \frac{1}{999}$ and η_r is set as in Theorem 4. The experiment was then run 100 times.



(a) Losses for a single trial.

(b) Losses averaged over 100 trials.

Figure 2.6: Tracking dynamics using DFS and comparing individual models for directional motion for the experiment in Section 2.4.3. Only shown are DMD losses for motions which are true before or after $t = 550$ for clarity. Before $t = 550$ the upward motion dynamic model incurs small loss, where as after $t = 550$ the motion to the right does well, and DFS successfully tracks this change.

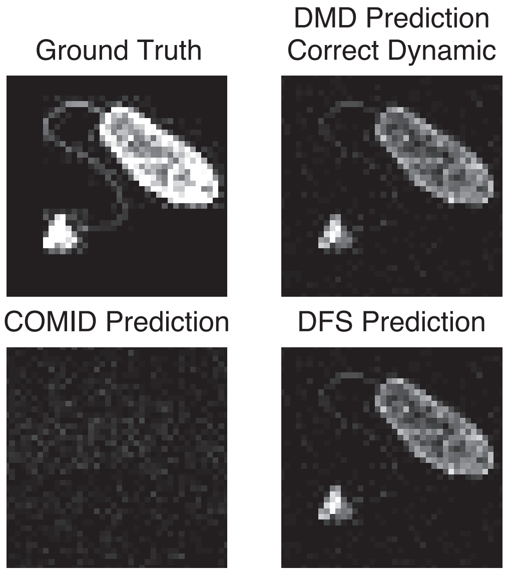


Figure 2.7: Dynamic compressed sensing experimental results. Zoomed in instantaneous predictions at $t = 1000$ for the experiment in Section 2.4.3. Top Left: θ_t. Top Right: $\widehat{\theta}_{\text{Right},t}$. Bottom Left: $\widehat{\theta}_{\text{COMID},t}$. Bottom Right: $\widehat{\theta}_t$. The prediction made with the prevailing motion is an accurate representation of the ground truth, while the prediction with the wrong dynamic is an unclear picture. The DFS algorithm correctly picks out the cleaner picture.

Figures 2.6 and 2.7 show the impact of using DFS. We see that DFS switches between dynamical models rapidly and outperforms all of the individual predictions, including COMID, used as a baseline, to show the advantages of incorporating knowledge of the dynamics.

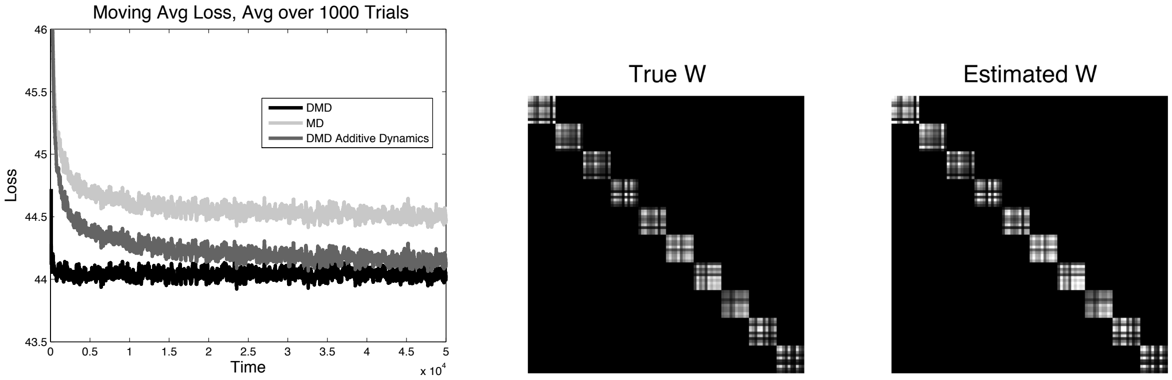
2.4.4 DMD with parametric additive dynamics: social network tracking

Finally, we look at self-exciting point processes on connected networks [10, 32]. Here we assume there is an underlying rate for nodes in a network which dictate how likely each node is to participate in an action. Then, based on which nodes act, it will increase other nodes likelihood to act in a dynamic fashion. For instance, in the context of social networks, we may observe events such as people meeting, corresponding, voting, or sharing information [42, 48, 50, 10, 59], and from this data wish to infer who's actions are influencing whose, and how those influences evolve over time. In a biological neural network, a node could correspond to a neuron and an action could correspond to a neural spike [13].

We simulate observations of a such a self-exciting point process in the following way:

$$\begin{aligned}\mu_{t+1} = \Phi_t(\mu_t, W) =& \tau\mu_t + W x_t + (1-\tau)\bar{\mu} \\ x_t \sim& \text{Poisson}(\mu_t)\end{aligned}$$

For our experiments $\mu_t \in (0,5]^{100}$ represents the average number of actions each of 100 nodes will make during time interval t, and $W \in [0,5]^{100\times 100}$ reflects the unknown underlying network structure which encodes how much an event by a one node will increase the likelihood of an event by another node in future time intervals. Here we assume τ is a known parameter between zero and one, $\bar{\mu} \in \mathbb{R}^{100}$ is a underlying base event rate.



(a) Moving average loss over previous 100 time points for DMD with a known W matrix, MD, and DMD exp averaged over 1000 trials.

(b) The true value and final estimate of W computed using Alg 3.

Figure 2.8: Experimental results tracking a self-exciting point process on a network, described in Section 2.4.4. Notice how the loss curve for Alg. 3 approaches the DMD curve (associated with clairvoyant knowledge of the underlying network matrix W) as the estimate of W improves, and significantly outperforms conventional mirror descent.

Our goal is to track the event rates μ_t and the network model W simultaneously; Algorithm 3 is applied with

$$\begin{aligned} \ell_t(\theta) =& \langle \mathbf{1}, \exp(\theta)\rangle - \langle x_t, \theta\rangle, & \tilde{\ell}_t(\mu) =& \langle \mathbf{1}, \mu\rangle - \langle x_t, \log\mu\rangle, \\ Z(\theta) =& \langle \mathbf{1}, \exp(\theta)\rangle, & \mu =& \nabla Z(\theta) = \exp(\theta). \end{aligned}$$

We generated data according to this model for $t = 1, ..., 50000$ for 1000 different trials, using $\tau = 0.5, \bar{\mu} = 0.1$ and W generated such that it is all zeros except on each distinct 10×10 block along the diagonal, elements are chosen to be uu^T for a vector $u \in [0.1, 1.1]^{10}$ with elements chosen uniformly at random. The matrix W is then normalized so that its spectral norm is 0.25 for stability. Using this generated data we ran DMD with known W (Alg. 1), MD, and DMD with additive dynamics (Alg. 3) to learn the dynamic rates. The step size parameters were set as $\eta_t = .9/\sqrt{t}$ and $\rho_t = .005/\sqrt{t}$. The results are shown for DMD with the matrix W known in advance, MD and Alg. 3 in Figure 2.8.

We again see several important characteristics in these plots. The first is that by incorporating knowledge of the dynamics, we incur significantly less loss than standard Mirror Descent. Secondly, we see that even without knowing what the values of the matrix W, we can learn it simultaneously with the rate vectors μ_t from streaming data, and the resulting accurate estimate leads to low loss in the estimates of the rates.

2.5 Conclusions and future directions

Processing high-velocity streams of high-dimensional data is a central challenge to big data analysis. Scientists and engineers continue to develop sensors capable of generating large quantities of data, but often only a small fraction of that data is carefully examined or analyzed. Fast algorithms for sifting through such data can help analysts track dynamic environments and identify important subsets of the data which are inconsistent with past observations.

In this paper we have proposed a novel online optimization method, called Dynamic Mirror Descent (DMD), which incorporates dynamical models into the prediction process and yields low regret bounds for broad classes of comparator sequences. The proposed methods are applicable for a wide variety of observation models, noise distributions, and dynamical models. There is no assumption within our analysis that there is a "true" known underlying dynamical model, or that the best dynamical model is unchanging with time. The proposed Dynamic Fixed Share (DFS) algorithm adaptively selects the most promising dynamical model from a family of candidates at each time step. Additionally we show methods which learn in parametric families of dynamical models. In experiments DMD shows strong tracking behavior even when underlying dynamical models are switching, in such applications as dynamic texture analysis, compressive video, and self-exciting point process analysis.

Bibliography

[1] D. Adamskiy, W. M. Koolen, A. Chernov, and V. Vovk. A closer look at adaptive regret. In *Proceedings of the 23rd international conference on Algorithmic Learning Theory*, ALT'12, pages 290–304, 2012.

[2] S. Amari and H. Nagaoka. *Methods of Information Geometry.* American Mathematical Society, Providence, 2000.

[3] D. Angelosante, J. A. Bazerque, and G. B. Giannakis. Online adaptive estimation of sparse signals: Where rls meets the 11-norm. *IEEE Transactions on Signal Processing*, 58(7):3436–3447, 2010.

[4] D. Angelosante, G. B. Giannakis, and E. Grossi. Compressed sensing of time-varying signals. In *Intl Conf. on Dig. Sig. Proc.*, 2009.

[5] K. S. Azoury and M. K. Warmuth. Relative loss bounds for on-line density estimation with the exponential family of distributions. *Machine Learning*, 43:211–246, 2001.

[6] A. Bain and D. Crisan. *Fundamentals of Stochastic Filtering.* Springer, 2009.

[7] O. Banerjee, L. El Ghaoui, and A. d'Aspremont. Model selection through sparse maximum likelihood estimation for multivariate Gaussian or binary data. *J. Mach. Learn. Res.*, 9:485–516, 2008.

[8] A. Beck and M. Teboulle. Mirror descent and nonlinear projected subgradient methods for convex programming. *Operations Research Letters*, 31:167–175, 2003.

[9] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Comput.*, 15(6):1373–1396, June 2003.

[10] C. Blundell, K. A. Heller, and J. M. Beck. Modelling reciprocating relationships with hawkes processes. In *Proc. NIPS*, 2012.

[11] S. Boyd and L. Vandenberghe. *Convex Optimization.* Cambrdige Univ. Press, Cambridge, UK, 2004.

[12] L. M. Bregman. The relaxation method of finding the common points of convex sets and its application to the solution of problems in convex programming. *Comput. Mathematics and Math. Phys.*, 7:200–217, 1967.

[13] Emery N Brown, Robert E Kass, and Partha P Mitra. Multiple neural spike train data analysis: state-of-the-art and future challenges. *Nature neuroscience*, 7(5):456–461, 2004.

[14] E. Candès, J. Romberg, and T. Tao. Stable signal recovery from incomplete and inaccurate measurements. *Communications on Pure and Applied Mathematics*, 59(8):1207–1223, 2006.

[15] Y. Censor and S. A. Zenios. *Parallel Optimization: Theory, Algorithms and Applications.* Oxford Univ. Press, Oxford, UK, 1997.

[16] N. Cesa-Bianchi, P. Gaillard, G. Lugosi, and G. Stoltz. A new look at shifting regret. arXiv:1202.3323, 2012.

[17] N. Cesa-Bianchi and G. Lugosi. *Prediction, Learning and Games.* Cambridge University Press, New York, 2006.

[18] W. Clavin. Managing the deluge of 'big data' from space, 2013. http://www.jpl.nasa.gov/news/news.php?release=2013-299.

[19] P. E. Dewdney, P. J. Hall, R. T. Schlilzzi, and T. J. L. W. Lazio. The square kilometre array. *Proceedings of the IEEE*, 97(8), 2009.

[20] G. Doretto, A. Chiuso, Y. Wu, and S. Soatto. Dynamic textures. *International Journal of Computer Vision*, 51(2):91–109, 2003.

[21] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single pixel imaging via compressive sampling. *IEEE Sig. Proc. Mag.*, 25(2):83–91, 2008.

[22] J. Duchi, S. Shalev-Shwartz, Y. Singer, and A. Tewari. Composite objective mirror descent. In *Conf. on Learning Theory (COLT)*, 2010.

[23] E. Dumbill. What is big data? An introduction to the big data landscape, 2012. http://strata.oreilly.com/2012/01/what-is-big-data.html.

[24] A. Gyorgy, T. Linder, and G. Lugosi. Efficient tracking of large classes of experts. *IEEE Transaction on Information Theory*, 58:6709–6725, November 2012.

[25] S. Haykin. *Adaptive Filter Theory.* Prentice-Hall, New Jersey, 2002.

[26] E. Hazan and C. Seshadhri. Efficient learning algorithms for changing environments. In *Proc. Int. Conf on Machine Learning (ICML)*, pages 393–400, 2009.

[27] M. Herbster and M. K. Warmuth. Tracking the best expert. *Machine Learning*, 32:151–178, 1998.

[28] M. Herbster and M. K. Warmuth. Tracking the best linear predictor. *Journal of Machine Learning Research*, 35(3):281–309, 2001.

[29] J.-B. Hiriart-Urruty and C. Lemaréchal. *Fundamentals of Convex Analysis.* Springer, Berlin, 2001.

[30] W.M. Koolen and S. de Rooij. Combining expert advice efficiently. In *Proceedgins of the 21st Annual Conference on Learning Theory (COLT 2008)*, pages 275–286, 2008.

[31] J. Langford, L. Li, and T. Zhang. Sparse online learning via truncated gradient. *J. Mach. Learn. Res.*, 10:777–801, 2009.

[32] Scott W. Linderman and Ryan P. Adams. Discovering latent network structure in point process data. arXiv:1402.0914, 2014.

[33] N. Littlestone and M. K. Warmuth. The weighted majority algorithm. *Inf. Comput.*, 108(2):212–261, 1994.

[34] R. Marcia and R. Willett. Compressive coded aperture video reconstruction. In *Proc. European Signal Processing Conference EUSIPCO*, 2008.

[35] J. Marlow. What to do with 1,000,000,000,000,000,000 bytes of astronomical data per day, 2012. Wired, http://www.wired.com/2012/04/what-to-do-with-1000000000000000000-bytes-of-astronomical-data-per-day/.

[36] J. Matson. Nasa readies a satellite to probe the sun–inside and out. *Scientific American*, 2010. `http://www.scientificamerican.com/article/solar-dynamics-observatory-sdo/`.

[37] B. McMahan. A unified view of regularized dual averaging and mirror descent with implicit updates. arXiv:1009.3240v2, 2011.

[38] N. Merhav and M. Feder. Universal prediction. *IEEE Trans. Info. Th.*, 44(6):2124–2147, October 1998.

[39] A. Nemirovski, A. Juditsky, G. Lan, and A. Shapiro. Robust stochastic approximation approach to stochastic programming. *SIAM J. Optim.*, 19(4):1574–1609, 2009.

[40] A. S. Nemirovsky and D. B. Yudin. *Problem complexity and method efficiency in optimization.* John Wiley & Sons, New York, 1983.

[41] J. Y. Park and M. B. Wakin. A multiscale framework for compressive sensing of video. In *Picture Coding Symposium (PCS)*, Chicago, IL, May 2009.

[42] M. Raginsky, R. Willett, C. Horn, J. Silva, and R. Marcia. Sequential anomaly detection in the presence of noise and limited feedback. *IEEE Transactions on Information Theory*, 58(8):5544–5562, 2012.

[43] A. Rakhlin and K. Sridharan. Online learning with predictable sequences. arXiv:1208.3728, 2012.

[44] Avinash Ravichandran, Rizwan Chaudhry, and Rene Vidal. Dynamic texture toolbox, 2011. `http://www.vision.jhu.edu`.

[45] P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimenstional Ising model selection using ℓ_1-regularized logistic regression. *Annals of Statistics*, 38:1287–1319, 2010.

[46] Aswin C. Sankaranarayanan, Christoph Studer, and Richard G. Baraniuk. CS-MUVI: video compressive sensing for spatial-multiplexing cameras. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, April 2012.

[47] C.R. Shalizi, A.Z. Jacobs, K.L. Klikner, and A Clauset. Adapting to non-stationarity with growing expert ensembles. arXiv:1103:0949, 2011.

[48] J. Silva and R. Willett. Hypergraph-based anomaly detection in very large networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(3):563–569, 2009. doi:10.1109/TPAMI.2008.232.

[49] T. A. B. Snijders. The statistical evaluation of social network dynamics. *Sociological Methodology*, 31(1):361–395, 2001.

[50] A. Stomakhin, M. B. Short, and A. Bertozzi. Reconstruction of missing data in social networks based on temporal patterns of interactions. *Inverse Problems*, 27(11), 2011.

[51] Vincent Studer, Jérome Bobin, Makhlad Chahid, Hamed Shams Mousavi, Emmanuel Candes, and Maxime Dahan. Compressive fluorescence microscopy for biological and hyperspectral imaging. *Proceedings of the National Academy of Sciences*, 109(26):E1679–E1687, 2012.

[52] M. Szummer and R. W. Picard. Temporal texture modeling. In *Proceedings of International Conference on Image Processing (ICIP)*, 1996.

[53] Y. Theodor and U. Shaked. Robust discrete-time minimum-variance filtering. *IEEE Trans. Sig. Proc.*, 44(2):181–189, 1996.

[54] N. Vaswani and W. Lu. Modified-CS: Modifying compressive sensing for problems with partially known support. *IEEE Trans. Sig. Proc.*, 58:4595–4607, 2010.

[55] V. Vovk. Aggregating algorithms. *Conf. on Learning Theory (COLT)*, 1990.

[56] M. J. Wainwright and M. I. Jordan. Graphical models, exponential families, and variational inference. *Foundations and Trends in Machine Learning*, 1(1–2):1–305, December 2008.

[57] L. Xiao. Dual averaging methods for regularized stochastic learning and online optimization. *J. Mach. Learn. Res.*, 11:2543–2596, 2010.

[58] L. Xie, Y. C. Soh, and C. E. de Souza. Robust Kalman filtering for uncertain discrete-time systems. *IEEE Trans. Autom. Control*, 39:1310–1314, 1994.

[59] K. Zhou, H. Zha, and L. Song. Learning social infectivity in sparse low-rank networks using multi-dimensional hawkes processes. In *Proceedings of the 16th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2013.

[60] M. Zinkevich. Online convex programming and generalized infinitesimal gradient descent. In *Proc. Int. Conf. on Machine Learning (ICML)*, pages 928–936, 2003.